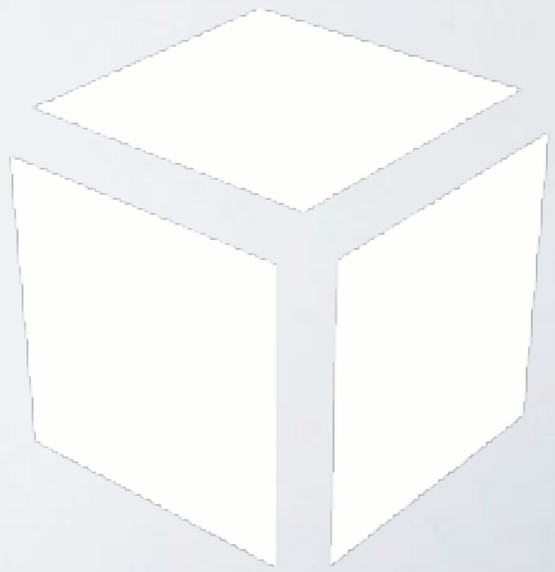
# Packaging Go Code in pkgsrc

Benny Siegert (bsiegert@NetBSD.org)

FOSDEM 2017

# pkgsrc
# https://pkgsrc.org/

- **The NetBSD packages collection.**
  - Over 17,000 packages!
- Runs on **23 platforms** (*BSD, Solaris, Linux, Windows).
- pkgin for binary packages
- can run unprivileged (no root)
- pkgsrc-wip: staging area, low entry barrier.



# Life of a package

- Makefile, distinfo, DESCR, PLIST
- Standard stages:

  fetch, checksum

  tools, depends

  extract, patch

  build

  stage-install

  package

  package-install

# Go

- Fairly popular programming language.
  - Go devroom @ H.1302 (Depage) tomorrow
- The **“go” tool** handles
  fetching,
  building,
  installing
  Go programs.

# go get, in action

```
$ export GOPATH=$PWD; go get -v github.com/ovh/go-ovh/...
github.com/ovh/go-ovh (download)
Fetching https://gopkg.in/ini.v1?go-get=1
Parsing meta tags from https://gopkg.in/ini.v1?go-get=1 (status code 200)
get "gopkg.in/ini.v1": found meta tag main.metaImport{Prefix:"gopkg.in/ini.v1", VCS:"git",
RepoRoot:"https://gopkg.in/ini.v1"} at https://gopkg.in/ini.v1?go-get=1
gopkg.in/ini.v1 (download)
gopkg.in/ini.v1
github.com/ovh/go-ovh/ovh
```

```
$ find src pkg -type d | fgrep -v .git
src
src/github.com
src/github.com/ovh
src/github.com/ovh/go-ovh
src/github.com/ovh/go-ovh/ovh
src/gopkg.in
src/gopkg.in/ini.v1
src/gopkg.in/ini.v1/testdata
pkg
pkg/darwin_amd64
pkg/darwin_amd64/github.com
pkg/darwin_amd64/github.com/ovh
pkg/darwin_amd64/github.com/ovh/go-ovh
pkg/darwin_amd64/gopkg.in
```

# Go and package systems

- Mismatch between the go tool and package managers.
  - Either: package only Go itself?
  - Or: Deal with it!

- Many small units, similar to Perl packages.
- Binaries are statically linked.
- pkgsrc has a framework: go-package.mk.

# Latest Release?

Personal Open source Business Explore Pricing Blog Support This repository Search Sign in Sign up

ovh / go-ovh

Watch 3 Star 27 Fork 9

Code Issues 0 Pull requests 0 Projects 0 Pulse Graphs

Releases Tags

# Makefile

```
# $NetBSD: Makefile,v 1.1 2017/01/24 17:22:09 bsiegert Exp $

DISTNAME=        go-ovh-0.20170102
CATEGORIES=      net
MASTER_SITES=    ${MASTER_SITE_GITHUB:=ovh/}
GITHUB_PROJECT=  go-ovh
GITHUB_TAG=      d2207178e10e4527e8f222fd8707982df8c3af17

MAINTAINER=      pkgsrc-users@NetBSD.org
HOMEPAGE=        https://${GO_SRCPATH}
COMMENT=         Lightweight Go wrapper around OVH APIs
LICENSE=         modified-bsd

WRKSRC=          ${WRKDIR}
GO_DIST_BASE=    ${GITHUB_PROJECT}-${GITHUB_TAG}
GO_SRCPATH=      github.com/ovh/go-ovh

.include "../../lang/go/go-package.mk"
.include "../../mk/bsd.pkg.mk"
```

# Trying to build

```
$ bmake
=> Bootstrap dependency digest>=20010302: found digest-20121220
===> Checking for vulnerabilities in go-ovh-0.20170102
=> Checksum SHA1 OK for go-ovh-0.20170102-d2207178e10e4527e8f222fd8707982df8c3af17.tar.gz
=> Checksum RMD160 OK for go-ovh-0.20170102-d2207178e10e4527e8f222fd8707982df8c3af17.tar.gz
=> Checksum SHA512 OK for go-ovh-0.20170102-d2207178e10e4527e8f222fd8707982df8c3af17.tar.gz
===> Installing dependencies for go-ovh-0.20170102
=> Tool dependency ccache-[0-9]*: found ccache-3.2.4
=> Tool dependency checkperms>=1.1: found checkperms-1.11
=> Build dependency go-1.7.4*: found go-1.7.4
=> Build dependency cwrappers>=20150314: found cwrappers-20160908
===> Overriding tools for go-ovh-0.20170102
===> Extracting for go-ovh-0.20170102
===> Patching for go-ovh-0.20170102
===> Creating toolchain wrappers for go-ovh-0.20170102
===> Configuring for go-ovh-0.20170102
=> Checking for portability problems in extracted files
===> Building for go-ovh-0.20170102
work/src/github.com/ovh/go-ovh/ovh/configuration.go:10:2: cannot find package "gopkg.in/ini.v1" in any of:
    /opt/pkg/go/src/gopkg.in/ini.v1 (from $GOROOT)
    /opt/pkgsrc/net/go-ovh/work/src/gopkg.in/ini.v1 (from $GOPATH)
    /opt/pkgsrc/net/go-ovh/work/.buildlink/gopkg/src/gopkg.in/ini.v1
*** Error code 1

Stop.
bmake[1]: stopped in /opt/pkgsrc/net/go-ovh
*** Error code 1
```

# Dependencies

- Let’s add the dependency:

```
.include "../../devel/go-ini/buildlink3.mk"
```

- Now building works!

```
===> Building for go-ovh-0.20170102
github.com/ovh/go-ovh/ovh
$
```

- What happened? A “shadow GOPATH”.

```
$ find work/.buildlink/gopkg/src -type d
work/.buildlink/gopkg/src
work/.buildlink/gopkg/src/gopkg.in
work/.buildlink/gopkg/src/gopkg.in/ini.v1
work/.buildlink/gopkg/src/gopkg.in/ini.v1/testdata
```

# go-ovh/buildlink3.mk

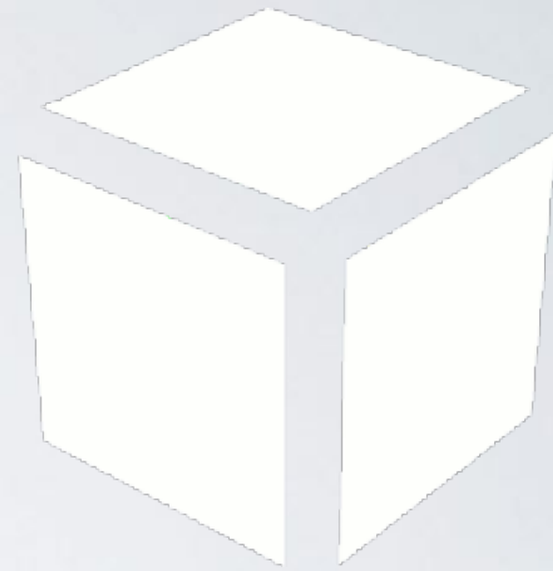
```
# $NetBSD: buildlink3.mk,v 1.1 2017/01/24 17:22:09 bsiegert Exp $

BUILDLINK_TREE+=        go-ovh

.if !defined(GO_OVH_BUILDLINK3_MK)
GO_OVH_BUILDLINK3_MK:=

BUILDLINK_CONTENTS_FILTER.go-ovh=       ${EGREP} gopkg/
BUILDLINK_DEPMETHOD.go-ovh?=            build

BUILDLINK_API_DEPENDS.go-ovh+=          go-ovh>=0.20170102
BUILDLINK_PKGSRCDIR.go-ovh?=            ../../net/go-ovh

.include "../../devel/go-ini/buildlink3.mk"
.endif  # GO_OVH_BUILDLINK3_MK

BUILDLINK_TREE+=        -go-ovh
```

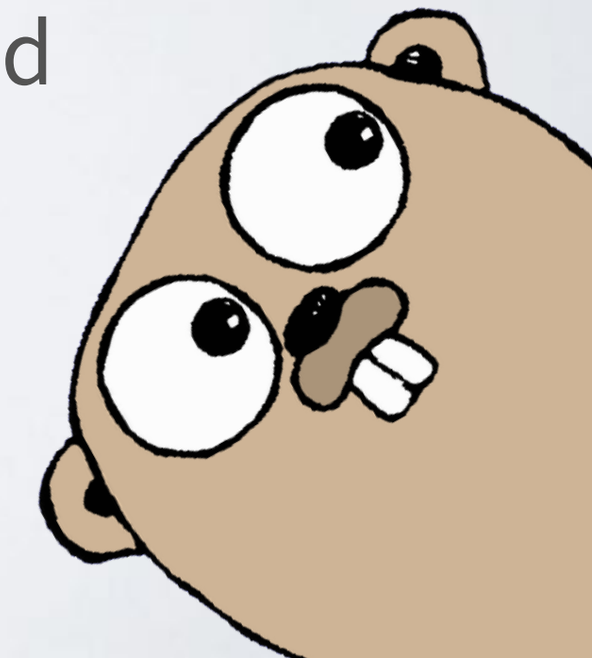
# This could be automated!

- Not a trivial task.
- Lots of metadata to discover:
  comment, license, release, long description
- Canonical import path vs. source code location
- Dependencies (test + build), sometimes C libraries
- I have previously written cpan2port, so …
  … maybe at some point?

# Tips for Upstream Authors

- Give those packagers some releases.
  Packagers love releases.
- Make your software “go get”able.
  - Antipattern: “use go run installer.go”
  - Antipattern: download stuff during the build
- Put a sensible description into the README.
- Avoid circular dependencies.





# Thank you!

Links:

https://fosdem.org/2017/schedule/event/packaging_go

https://pkgsrc.org/

https://pkgsrc.se/

https://golang.org/